

Peter Recktenwald

Hacks für die Digitale Fotografie

Digitalfotografie mit Arduino
und Raspberry Pi



Inhaltsverzeichnis

Vorwort	vii
1. Basic Hacks	1
Hack 01. Elektronik für Dummies	2
Hack 02. Grundlagen der C/C++-Programmierung	17
Hack 03. Erste Schritte mit Arduino	35
Hack 04. Arduino – Belichtungsmesser	51
Hack 05. Crashkurs in Java	61
Hack 06. Raspberry Pi – Einstieg	70
Hack 07. Raspberry Pi-DigiCam	82
2. Nützliche Tools	89
Hack 08. Einstieg in Fritzing	89
Hack 09. 2D/3D-Design leicht gemacht	101
Hack 10. Windows-Tools für den Raspberry Pi	113
Hack 11. Hilfe, meine Daten sind weg	115
Hack 12. Fotoaufnahmetisch	117
Hack 13. Stativ-Halterung für den Raspberry Pi	125
Hack 14. Kamera-Slider	128
Hack 15. Fotozubehör	130
Hack 16. Werkzeug und Verbrauchsmaterialien	133
3. Hacks zum Bau eines Foto-Controllers	143
Hack 17. Foto-Shield	144
Hack 18. Mini-Foto-Control	150
Hack 19. Modul zur Kamera-Fernsteuerung	160
Hack 20. Modul zur Blitzfernsteuerung	171
Hack 21. Ausgabemodul	176
Hack 22. Joystick-Modul	184
Hack 23. Analoger Joystick	192
Hack 24. Drehgeber-Modul	199

Hack 25. Bluetooth-Modul	209
Hack 26. RJ11-I ² C-Modul	213
Hack 27. Gehäuse für den Foto-Controller	221
Hack 28. Die Kameraauslöseverzögerung messen	230
4. Hacks zur Erweiterung des Foto-Controllers	235
Hack 29. FET-Modul	235
Hack 30. Infrarot-Fernsteuerung	241
Hack 31. Lichtschranke	245
Hack 32. Schall-Sensor	250
Hack 33. Lichtsensor	253
Hack 34. Bewegungsmelder	255
Hack 35. Servomotor-Steuerung	259
Hack 36. Schrittmotor-Steuerung	265
Hack 37. Getriebemotor-Steuerung	277
5. Hacks für Zeitraffer- und Highspeed-Fotografie	285
Hack 38. Zeitrafferaufnahmen	291
Hack 39. Tropfen auf Tropfen – Tropfenfotografie	295
Hack 40. Kurzzeitfotografie mit Sensor-Trigger	304
Hack 41. Gewitterblitz-Fotografie	308
Hack 42. Makro – Focus Stacking	310
Hack 43. Alles dreht sich auf dem Drehteller	318
Hack 44. Motorisierter Kamera-Slider	325
Hack 45. Malen mit Licht	330
Hack 46. Funkfernsteuerung	338
6. Raspberry Pi-Hacks für die Digitale Fotografie	343
Hack 47. Raspberry Pi Schnappschuss Kamera	343
Hack 48. In die (Foto-)Falle gegangen	352
Hack 49. Panorama Steuerung	356
Hack 50. USB-Kamerasteuerung 1	368
Hack 51. USB Kamera Steuerung 2	369
Index	371

Datensicherung

Klar, jeder erstellt ein Backup seiner Daten, insbesondere auch seiner Fotos und Videos, das ist doch selbstverständlich – schließlich lebt eine Festplatte nicht ewig. Aber was ist mit den Daten auf der Flash-Karte meiner Kamera? Auch diese kann jederzeit das Zeitliche segnen und ihren Inhalt mit in das digitale Nirvana nehmen.

Fotos und Filme von einer beschädigten Flash-Karte wiederherstellen

Das Unvorstellbare ist geschehen. Die Flash-Karte der Kamera ist plötzlich nicht mehr lesbar – und das ausgerechnet nach einer ausgiebigen Foto-Session. Vielleicht sind einzigartige Aufnahmen, die sich nicht wiederholen lassen, für immer verloren. Das wäre schon schlimm, aber es besteht Hoffnung, die Daten doch noch zu retten, ohne einen teuren Datenrettungsdienst in Anspruch nehmen zu müssen. Mir selbst ist ein solches Unglück bisher nur einmal geschehen.

Um es vorwegzunehmen – mit den Boardmitteln von Windows wie `Chkdsk` ist hier nichts zu holen. Im Gegenteil, jedwede Reparaturversuche werden nur dazu führen, dass die Daten anschließend restlos verloren sind. Auch andere Tools zur Wiederherstellung von Daten, die versuchen, die Daten anhand der FAT-Tabelle zu retten, sind hier überfordert. In solchen Fällen helfen nur noch Spezial-Tools.

In meinem konkreten Fall konnte ich mit Hilfe von *PhotoRec* (<http://www.cgsecurity.org/wiki/PhotoRec>) alle Daten erfolgreich wiederherstellen. Was aber macht dieses Tool anders? Es umgeht das Dateisystem und kann anhand der Datensektoren auf dem Medium viele Dateiformate, darunter Fotos und Videos, wiederherstellen. Dabei wird von dem defekten Medium nur gelesen, niemals darauf geschrieben.

PhotoRec läuft unter den gängigen Betriebssystemen (Mac-OSX, Linux, Windows) und muss nicht installiert werden. Die heruntergeladene Datei im Bundle mit dem Tool TestDisk wird lediglich in einem beliebigen Ordner entpackt. Gestartet wird PhotoRec unter Windows mit einem Doppelklick auf `photorec_win`.

PhotoRec 6.14, Data Recovery Utility, July 2013
 Christophe GRENIER <grenier@cgsecurity.org>
<http://www.cgsecurity.org>

PhotoRec is free software, and
 comes with ABSOLUTELY NO WARRANTY.

```
Select a media (use Arrow keys, then press Enter):
>Disk /dev/sda - 160 GB / 149 GiB (R0) - INTEL SSDSA2M160G2GC
  Disk /dev/sdb - 1500 GB / 1397 GiB (R0) - SAMSUNG HD154UI
  Disk /dev/sdc - 2000 GB / 1863 GiB (R0) - LaCie d2 quadra
  Disk /dev/sdd - 16 GB / 14 GiB (R0) - Generic Compact Flash
```

```
>[Proceed ] [ Quit ]
```

Zugegeben, die Oberfläche wirkt spartanisch – genau genommen handelt es sich gar nicht um eine grafische Oberfläche, sondern nur um ein Kommandozeilen-Tool. Die Bedienung ist jedoch einfach und intuitiv. Zunächst wird das richtige Quell-Laufwerk ausgewählt, weiter geht es dann mit **Proceed**.

PhotoRec 6.14, Data Recovery Utility, July 2013
Christophe GRENIER <grenier@cgsecurity.org>
<http://www.cgsecurity.org>

Disk /dev/sdd - 16 GB / 14 GiB (R0) - Generic Compact Flash

Partition	Start	End	Size in sectors
No partition	0 0 1	1946 233 63	31277232 [Whole disk]
> 1 P FAT32 LBA	0 1 1	1946 233 63	31277169 [EOS_DIGITAL]

>[Search] [Options] [File Opt] [Quit]
Start file recovery

Besondere Optionen sind normalerweise nicht nötig. Mit **Search** geht es anschließend weiter.

PhotoRec 6.14, Data Recovery Utility, July 2013
Christophe GRENIER <grenier@cgsecurity.org>
<http://www.cgsecurity.org>

1 P FAT32 LBA 0 1 1 1946 233 63 31277169 [EOS_DIGITAL]

Please choose if all space need to be analysed:
[Free] Scan for file from FAT32 unallocated space only
>[Whole] Extract files from whole partition

Der Menüpunkt **Whole** sollte angewählt werden, es sei denn, es wurde unabsichtlich die Partition gelöscht.

PhotoRec 6.14, Data Recovery Utility, July 2013

Please select a destination to save the recovered files.
Do not choose to write the files to the same partition they were stored on.
Keys: Arrow keys to select another directory
C when the destination is correct
Q to quit

Als Nächstes wird das Zielverzeichnis ausgewählt, in das die wiederhergestellten Dateien kopiert werden sollen. Mit der Taste C wird dann die Wiederherstellung gestartet.

PhotoRec 6.14, Data Recovery Utility, July 2013
Christophe GRENIER <grenier@cgsecurity.org>
<http://www.cgsecurity.org>

Disk /dev/sdd - 16 GB / 14 GiB (R0) - Generic Compact Flash

Partition	Start	End	Size in sectors
1 P FAT32 LBA	0 1 1	1946 233 63	31277169 [EOS_DIGITAL]

116 files saved in /cygdrive/f/Users/Peter/Downloads/Recovery/recup_dir director
Recovery completed.

[Quit]

Es dauert eine Weile, bis das Programm die Wiederherstellung abgeschlossen hat. Anschließend sollte soweit alles wieder hergestellt sein, was noch zu retten war. Es sind zwar alle Dateinamen verlorengegangen, aber das ist das geringste Problem. Die Flash-Karte kann anschließend neu formatiert und wieder verwendet werden.

HACK 12 Fotoaufnahmetisch

Der Aufnahmetisch dient zur Fotografie von kleineren Objekten. Der hier vorgestellte Aufnahmetisch kann unter anderem auch für folgende Zwecke genutzt werden:

- Tropfenfotografie (siehe Abbildung 5-3)
- Tabletop-Fotografie

Mit wenigen Handgriffen kann von einer Anwendung zur nächsten gewechselt werden. Als Beispiel soll uns im Folgenden der Tabletop-Aufbau dienen. Praktisch alle Aufnahmen für dieses Buch wurden damit aufgenommen. Zum Aufnahmetisch ge-

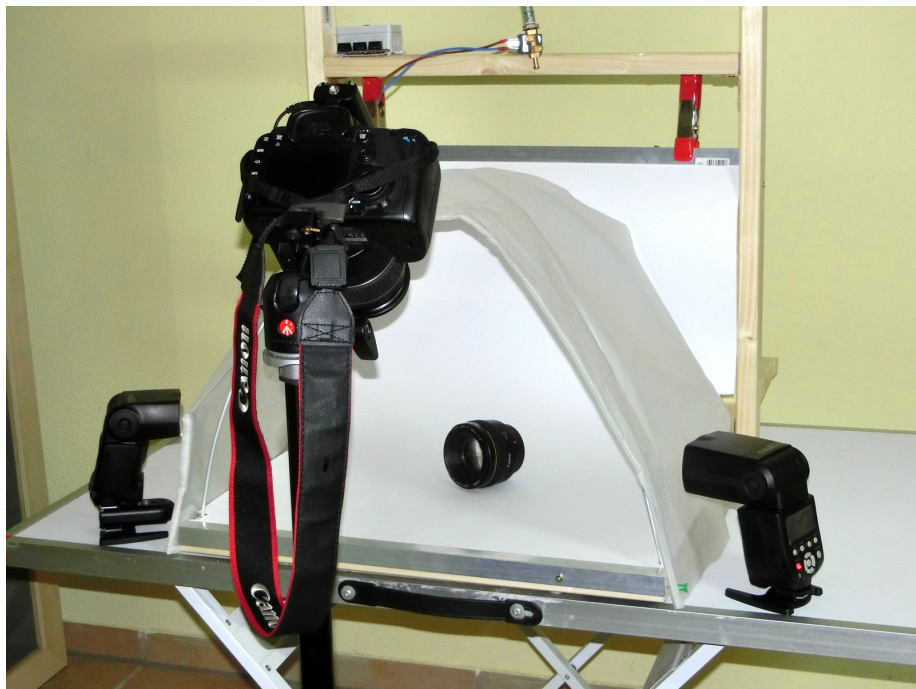


Abbildung 2-45: Aufnahmetisch – Tabletop

Foto-Setup

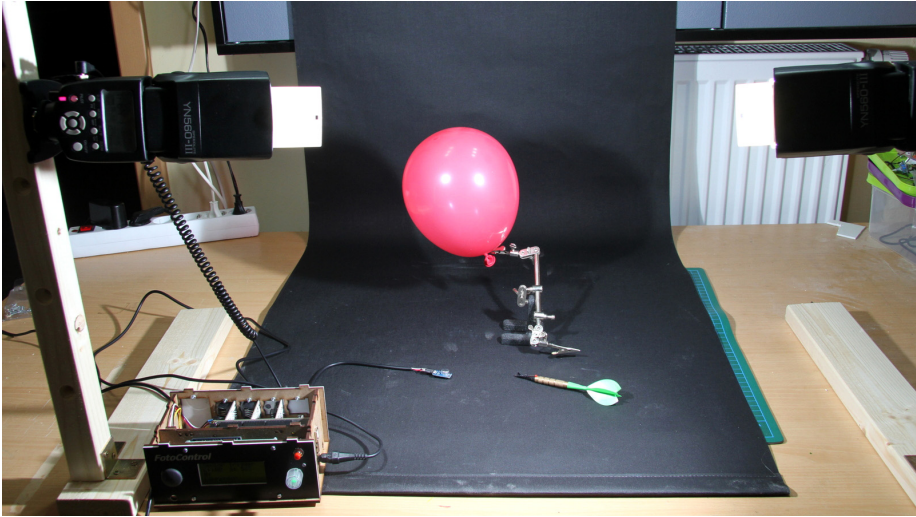


Abbildung 5-10: Sensor-Trigger – Foto-Setup

HACK 41

Gewitterblitz-Fotografie

Eine Variante der ereignisgesteuerten Aufnahme ist die Gewitterblitz-Fotografie. Die Aufnahme von Gewitterblitzen erfordert sehr viel Glück oder eine automatisierte Methode mit einem geeigneten Sensor. Ein Schallsensor wäre hier keine gute Wahl. Der Schall des Donners kommt sehr viel später beim Beobachter an als das Licht. Die *Schallgeschwindigkeit* (<http://de.wikipedia.org/wiki/Schallgeschwindigkeit>) liegt bei 340 m/s, während sich *Licht* (<http://de.wikipedia.org/wiki/Lichtgeschwindigkeit>) mit sagenhaften 300000 m/s fortbewegt. Also muss ein Lichtsensor her.

Aber ist das ausreichend? Wie lange dauert ein solcher Blitz? Wird die Kamera schnell genug ausgelöst? Die Zeitdauer eines Blitzes inklusive diverser Vorblitze liegt bei 100 bis 250ms. Das liegt schon im Bereich der Kameraauslöseverzögerung. Der eigentliche Hauptblitz dauert nur ca. 10ms. Bei den Vorblitzen ist aber eventuell das Licht nicht ausreichend. Dies sind eigentlich keine guten Voraussetzungen, um eine erfolgreiche Aufnahme zu erhalten. Man kann nur auf das Auftreten von mehreren Blitzen warten und die Kamera mit dem ersten Blitz auslösen, um dann den Verschluss länger offen zu lassen, in der Hoffnung, den ein oder anderen Nachfolgeblick zu erwischen.



Abbildung 5-11: Gewitterblitz

Warnung, Lebensgefahr!

Es ist sicher keine gute Idee, sich bei einem Gewitter im Freien aufzuhalten. Es besteht Lebensgefahr! Sie sollten sich auf jedem Fall in ein Gebäude oder ein Fahrzeug (Faradayscher Käfig (http://de.wikipedia.org/wiki/Faradayscher_K%C3%A4fig)) zurückziehen.

Was wird benötigt

- Arduino mit Fotoshield aus Hack #17
- alternativ Mini-Foto-Controller aus Hack #18
- Modul für die Kamerafernsteuerung Hack #19
- Lichtsensor (siehe Hack #33)
- PC/Laptop zum Einstellen und Anzeigen der Einstellungen über USB

Optional zur direkten Eingabe und Anzeige von Einstellungen:

- Drehgeber-Modul aus Hack #24 bzw. Joystick-Modul aus Hack #22
- Ausgabe-Modul aus Hack #21

An Foto-Equipment wird Folgendes benötigt:

- Kamera mit Stativ

Software

Es wird das leicht abgewandelte Sensor-Trigger-Programm verwendet, aus Hack #40. Es unterscheidet sich lediglich in der Funktion `runSensorTrigger()`. Statt den Blitz auszulösen, wird die Kamera direkt ausgelöst und eine Langzeitbelichtung gestartet, sobald der Sensor-Schwellwert erreicht wurde.

Beispiel 5-7: Gewitterblitz-Programm

```
...
int runSensorTrigger(void)
{
    int ret = 0;

    readSensors();           // Lese Sensor
    if (sensorValue < levelValue) // Schwellwert erreicht?
    {
        myCam.launchFocusShutter(); // Kamera Ausloeser druecken
        delay(delayTime);           // warten
        myCam.releaseFocusShutter(); // Kamera Ausloeser loslassen
        ret = 1;
    }
    return ret;
}
...
```

HACK 42 Makro – Focus Stacking

Fokus Stacking ist eine besondere Technik, bei der mehrere Aufnahmen mit unterschiedlichem Fokus zu einer Aufnahme zusammengefügt wird. Diese Technik ist insbesondere für Makro-Fotografie interessant, weil hier die Tiefenschärfe besonders gering ist. Durch das Fokus Stacking erhält man nach der Bildbearbeitung eine durchweg scharfe Abbildung des Fotomotivs.

Für das Fokus Stacking gibt es im Prinzip zwei Möglichkeiten:

- Der Abstand zum Motiv bleibt unverändert. Die Fokus-Einstellung am Kamera Objektiv wird von Aufnahme zu Aufnahme geändert.
- Die Fokus-Einstellung am Kamera-Objektiv bleibt unverändert. Der Abstand zum Motiv wird von Aufnahme zu Aufnahme geändert.

Wir werden hier die zweite Methode wählen und eine Makroschiene verwenden, um den Motivabstand zu ändern. Natürlich kommt wieder unser Foto-Controller zum Einsatz, um den Vorgang zu automatisieren. Dazu muss die Makroschiene mit einem Motor, genauer gesagt mit einem Schrittmotor angesteuert werden.

```

// create and register gpio pin listener
joyDBtn.addListener(new GpioPinListenerDigital() {
    @Override
    public void handleGpioPinDigitalStateChangeEvent(GpioPinDigitalState
                                                    ChangeEvent event) {

        // display pin state on console
        System.out.println(" --> Joystick D Button: " + event.getPin()
                            + " = " + event.getState());

        //if (event.getState().equals("HIGH"))
            mode = eModes.MOD_VIDEO;
    }

});

// create and register gpio pin listener
joyCenterBtn.addListener(new GpioPinListenerDigital() {
    @Override
    public void handleGpioPinDigitalStateChangeEvent(GpioPinDigitalState
                                                    ChangeEvent event) {

        // display pin state on console
        System.out.println(" --> Joystick Center Button: "
                            + event.getPin() + " = " + event.getState());

        ledPin.low();
        piCam.shutdown();
        Runtime.getRuntime().exit(0);
    }

});

System.out.println("...see the listener feedback here in the console.");

// keep program running until user aborts (CTRL-C)
for (;;) {
    Thread.sleep(500);
}
}
}

```

HACK 48

In die (Foto-)Falle gegangen

Eine Fotofalle kann man mit verschiedenen Sensoren verwirklichen, z. B. Mikrofon, Lichtschranken oder Bewegungsmelder. Da in unserem Garten gerade ein Blaumeisenpärchen brütet, fiel hier die Entscheidung für einen Bewegungsmelder. Die Vögel ließen sich zum Glück nicht durch die Aufbauarbeiten stören.

Ganz so einfach war die Durchführung dann natürlich trotzdem nicht. Der Bewegungsmelder hat einen sehr weiten Erfassungswinkel von 110°. Also musste eine Ab-

deckung um den Sensor montiert werden, um den Erfassungswinkel einzuschränken. Trotzdem gab es immer noch viele Fehlauslösungen. Und selbst wenn einer der Vögel den Sensor auslöste, hieß das noch lange nicht, dass das Foto auch brauchbar war. Aber der Super-GAU passierte an dem Tag, an dem die Jungvögel ihr Nest verließen. Ausgerechnet an diesem Tag regnete es und ich musste die ungeschützte Kamera abbauen. Wie ich lernen musste, schlüpfen Blaumeisenjungen häufig alle an einem Tag und kehren dann auch nicht wieder in ihr Nest zurück. Pech gehabt. Nächstes Jahr wird aufgerüstet, mit einer PiNoIR Camera im Vogelhaus und einer wettergeschützten zweiten Kamera draußen.

Auch unser Futterhaus war diesen Winter wieder ständig belagert von Futter suchenden Vögeln.



Abbildung 6-6: Raspberry Pi Fotofalle Futterstelle

Was wird benötigt?

- Raspberry Pi Board
- Raspberry Pi Kamera
- Bewegungsmelder, siehe Hack #34
- Halterung für Raspberry Pi und Kamera, siehe Hack #13
- Foto Stativ
- Wechselplatte, passend zum Fotostativ

Verdrahtung

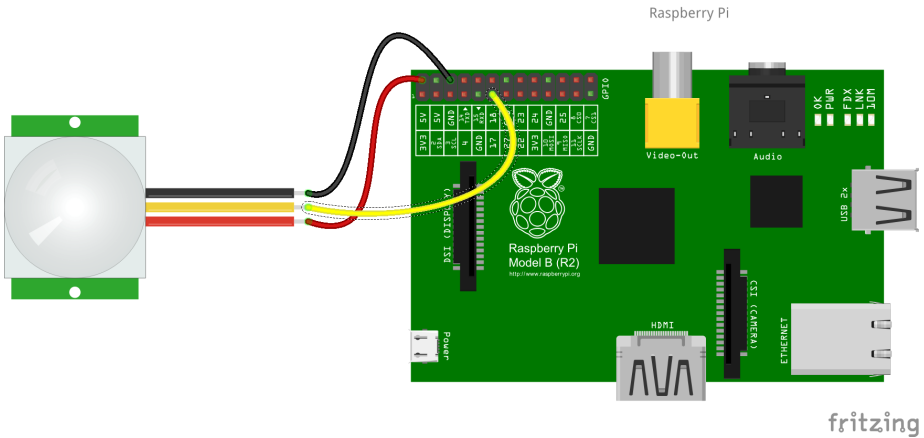


Abbildung 6-7: Raspberry Pi Fotofalle Verdrahtung

Software

Die Software für die Fotofalle wurde in Java geschrieben und benutzt die bereits bekannte **Camera**-Klasse und die **Pi4J**-Bibliothek.

Das Programm wird unter Netbeans übersetzt. Im Ordner **Dist** befinden sich das übersetzte jar-Archiv und die benötigten Bibliotheken. Den Inhalt des **Dist**-Ordners kopiert man über SCP auf den Raspberry Pi in einen eigenen Ordner mit dem Namen **RpiCameraTrap**.

Damit das Programm nach dem Einschalten direkt gestartet werden kann, wird die Datei **/etc/rc.local** angepasst:

```
sudo nano /etc/rc.local
```

Vor die Zeile **exit 0** fügt man den Befehl zum Start der Kameraanwendung ein:

```
cd /home/pi/RpiCameraTrap
sudo /opt/jdk1.8.0/bin/java -jar ./RpiCameraTrap.jar
```

Nach einem Neustart startet das Programm dann direkt ohne Anmeldung am Raspberry Pi.

Beispiel 6-2: Raspberry Pi Foto Falle

```
/* Digital Fotografie Hacks
 * Raspberry Pi Foto Falle:
 * RPiCameraTrap Klasse
 * v0.1, 2014-07-02, Peter Recktenwald
 */
```

```

public class RPiCameraTrap {

public static String GetActDate() {
    DateFormat df = new SimpleDateFormat("yyyy-MM-dd-HH-mm-ss");

    // Get the date today using Calendar object.
    Date today = Calendar.getInstance().getTime();
    // Using DateFormat format method we can create a string
    // representation of a date with the defined format.
    String reportDate = df.format(today);
    return reportDate;
}

/**
 * @param args the command line arguments
 */
public static void main(String[] args) throws InterruptedException {
    // TODO code application logic here
    Camera piCam = new Camera();
    // create gpio controller
    final GpioController gpio = GpioFactory.getInstance();

    // provision gpio pin #00 as an input pin with its internal pull
    // up resistor enabled
    final GpioPinDigitalInput myButton
= gpio.provisionDigitalInputPin(RaspiPin.GPIO_00, PinPullResistance.PULL_UP);

    // create and register gpio pin listener
    myButton.addListener(new GpioPinListenerDigital() {
        @Override
        public void handleGpioPinDigitalStateChangeEvent(GpioPinDigitalState
ChangeEvent event) {

            // display pin state on console
            System.out.println(" --> GPIO PIN STATE CHANGE: "
+ event.getPin() + " = " + event.getState());
            piCam.takePicture("pic" + GetActDate());
        }
    });

    System.out.println(" ... complete the GPIO #00 circuit and see the
listener feedback here in the console.");

    // keep program running until user aborts (CTRL-C)
    for (;;) {
        Thread.sleep(500);
    }
}
}

```